

# Enforcing Safety of Real-Time Schedules on Contemporary Processors using a Virtual Simple Architecture (VISA)

**Aravindh Anantaraman**<sup>\*</sup>, Kiran Seth<sup>†</sup>,  
Eric Rotenberg<sup>\*</sup>, Frank Mueller<sup>‡</sup>



Center for Embedded Systems Research (CESR)

<sup>\*</sup>Electrical & Computer Eng./ <sup>‡</sup> Computer Science

North Carolina State University

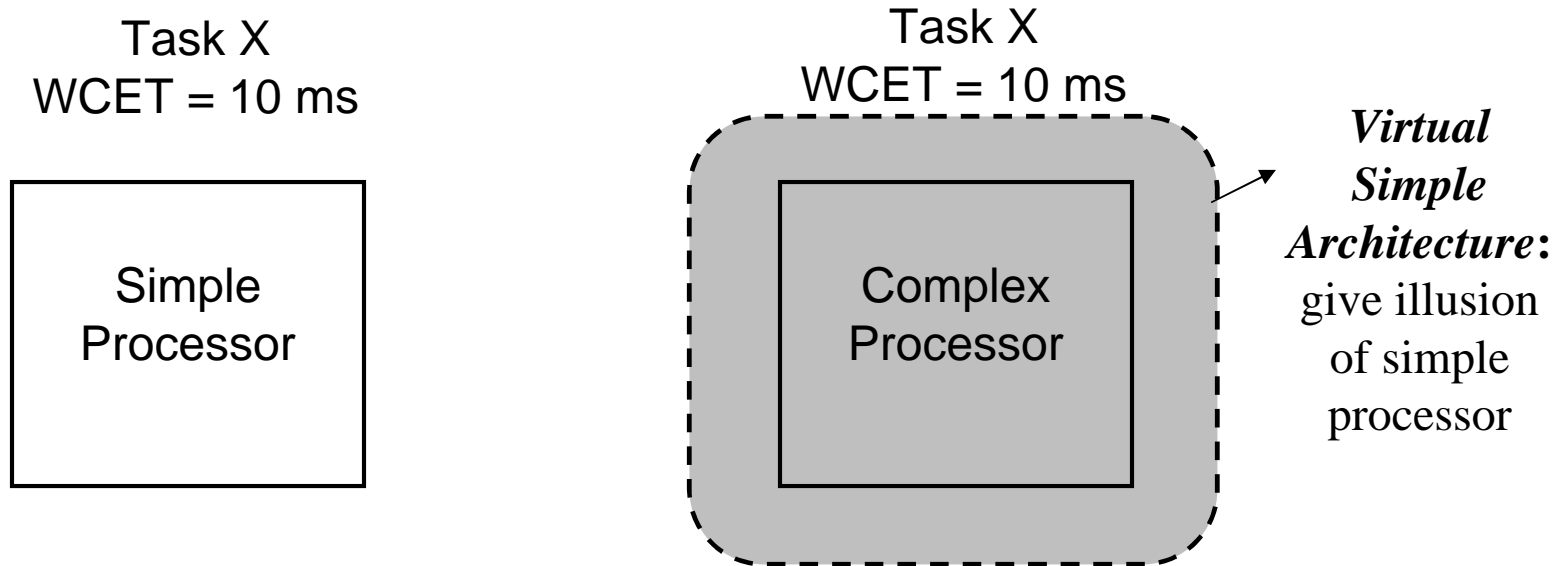
<sup>†</sup> Qualcomm. Inc



# Complexity in Hard-Real-Time Systems

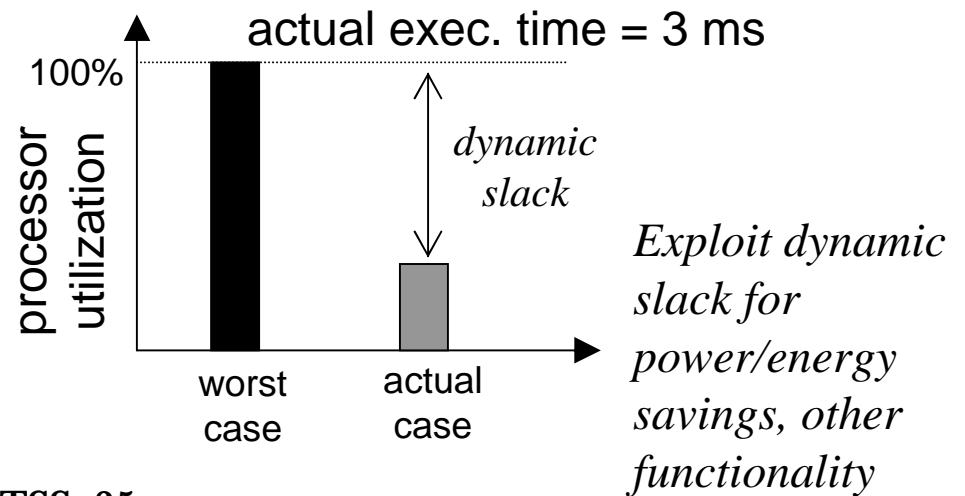
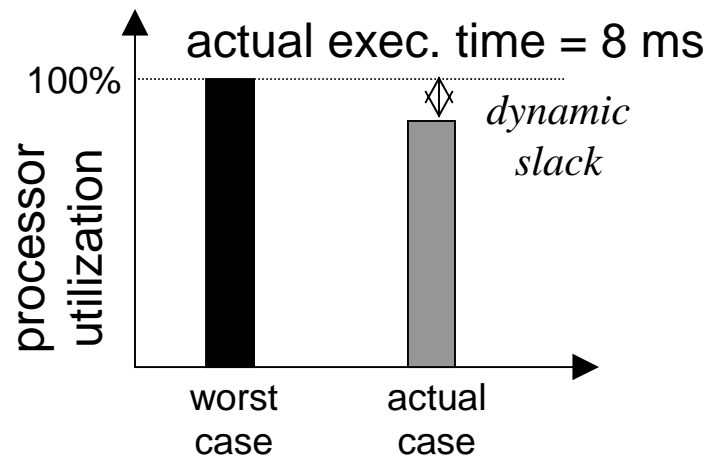
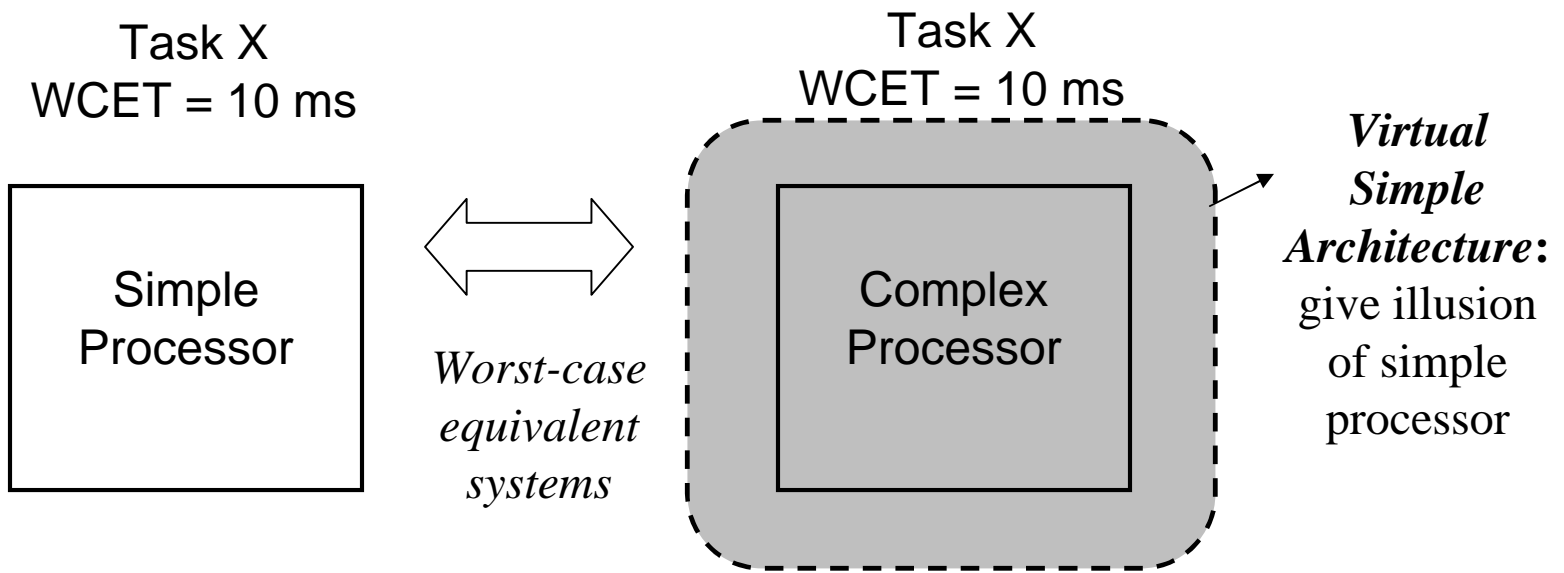
- Worst-case execution time (WCET) crucial for schedulability analysis
- Contemporary processors are extremely complex
  - Branch prediction, pipelining, out-of-order execution
  - Improve average case performance
  - WCET unknown
- Complex processors not used in real-time systems

# Virtual Simple Architecture (VISA)



- Novel non-literal approach to static timing analysis
  - Use simple processor as proxy for complex processor
  - *Dynamically* guarantee WCET

# Virtual Simple Architecture (VISA)



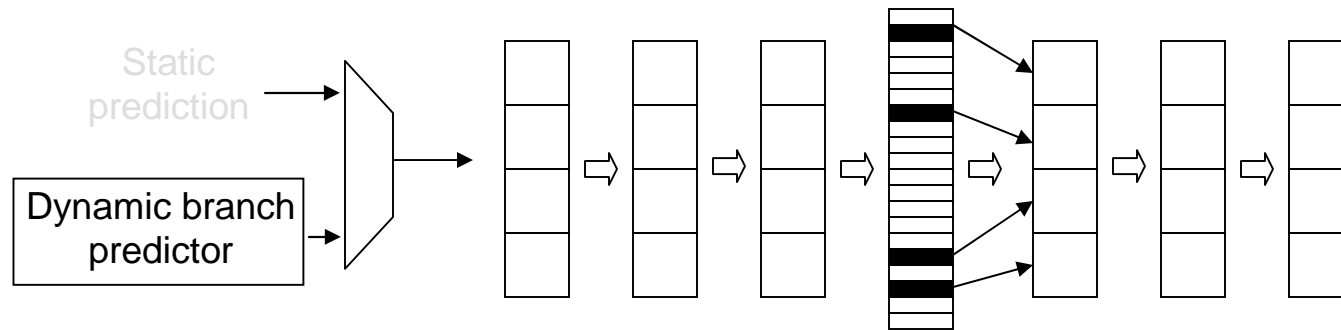
# Previous Approaches

1. **Avoid complexity**
  - VISA allows complex processors to be used
2. **Disable complexity during hard-real-time tasks**
  - VISA disables complexity only when problematic
3. **Continue research in timing analysis**
  - WCET of simple proxy improved

# VISA Overview

- Provides real-time guarantees for contemporary processors
- Approach
  - Execute tasks optimistically on complex mode
  - Gauge interim progress
  - Safe back-up mode for anomalous scenarios

# Dual-Mode VISA Processor



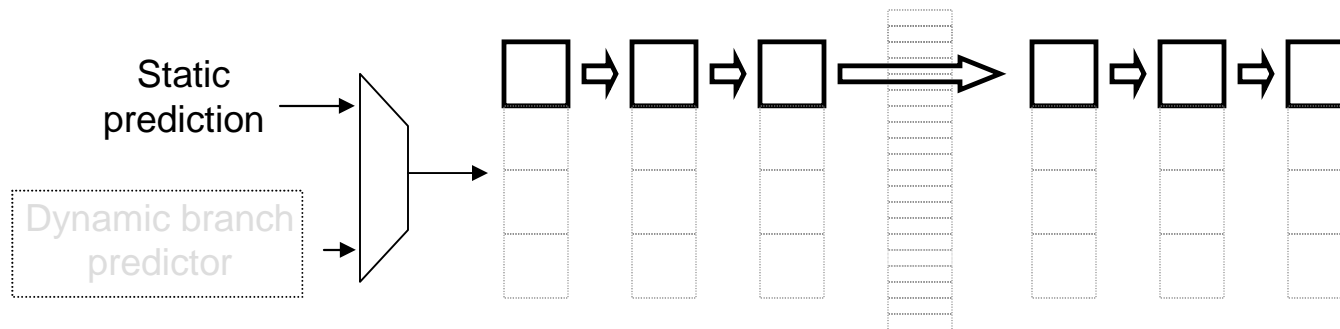
## Complex mode

- dynamic branch prediction
- superscalar
- out-of-order execution

## Simple mode

- static branch prediction
- scalar
- in-order execution

# Dual-Mode VISA Processor



## Complex mode

- dynamic branch prediction
- superscalar
- out-of-order execution

## Simple mode

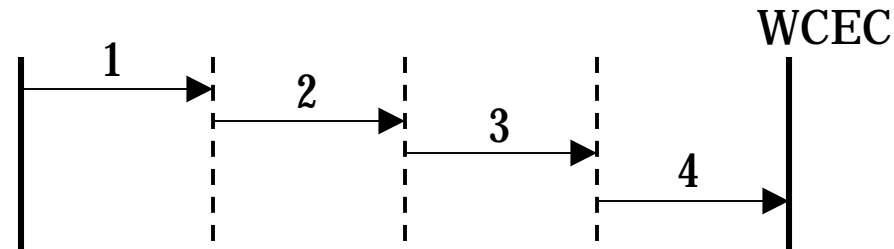
- static branch prediction
- scalar
- in-order execution



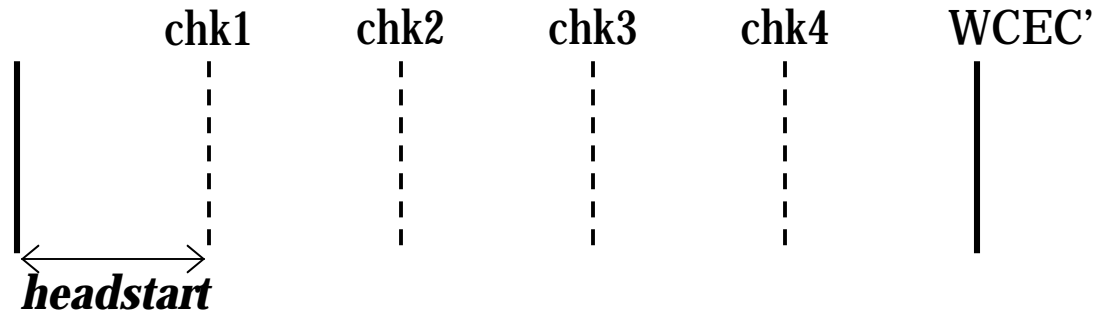
# VISA in Action

simple mode  $\longrightarrow$   
 complex mode  $\dashrightarrow$

**Non-speculative  
simple mode**



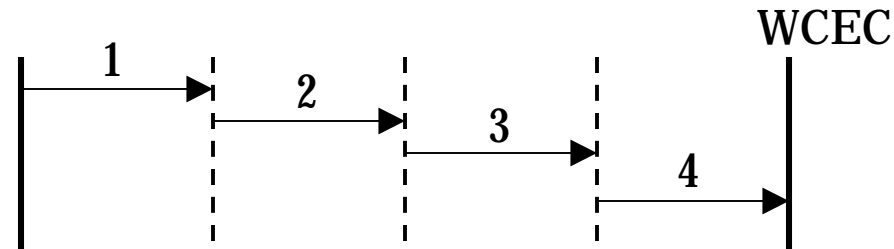
**Successful  
speculation in  
complex mode**



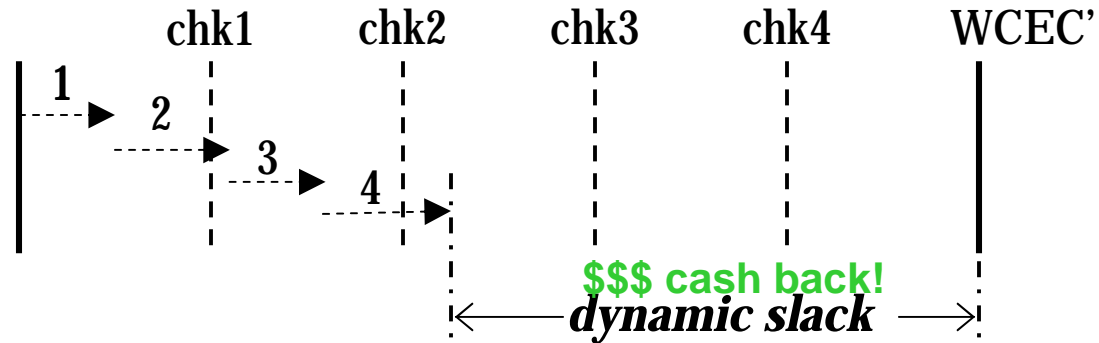
# VISA in Action

simple mode  $\longrightarrow$   
 complex mode  $\dashrightarrow$

**Non-speculative  
simple mode**



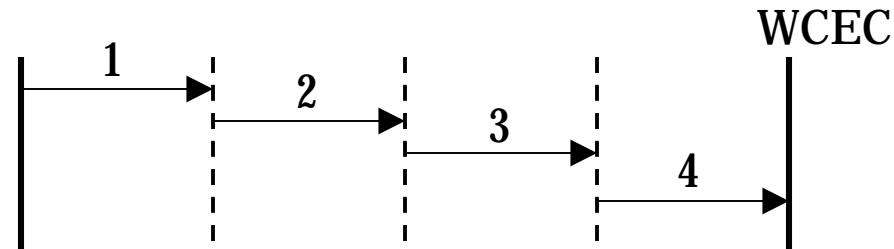
**Successful  
speculation in  
complex mode**



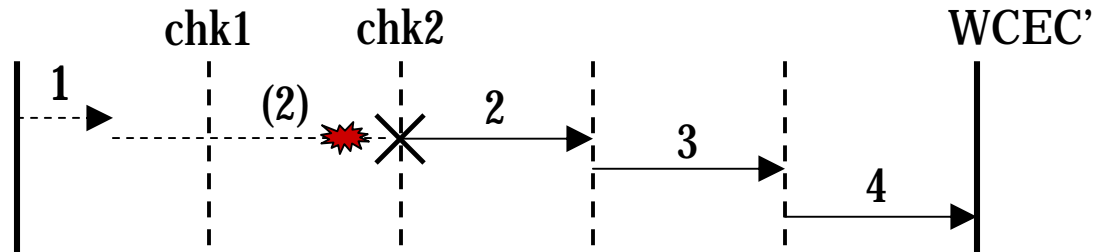
# VISA in Action

simple mode  $\longrightarrow$   
 complex mode  $\dashrightarrow$

**Non-speculative  
 simple mode**



**Misspeculation  
 in complex mode**

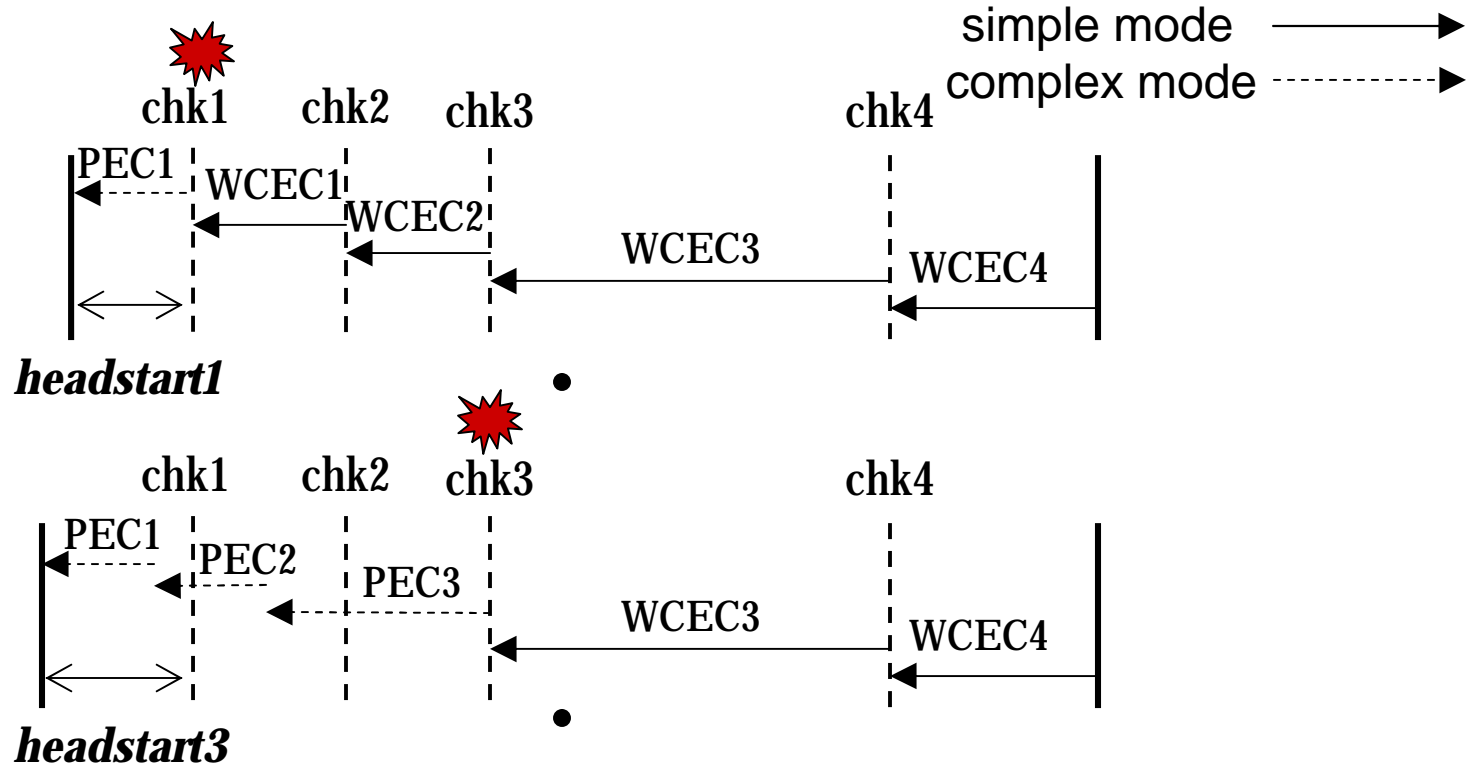


***WCET  
 preserved in  
 spite of missed  
 checkpoint***

# Contributions

- Minimize headstart overhead
- Novel zero-overhead VISA approach – *dynamic headstart accrual*
- Extend VISA to multi-tasking systems
- Energy evaluation in multi-tasking systems

# Headstart Assessment



$$headstart_k = \left( \sum_{i=1}^{k-1} PEC_i + PEC_k + WCEC_k + \sum_{i=k+1}^s WCEC_i \right) - WCEC$$

# Explicit Padding Approach

- Pad task WCEC with max headstart amount
- Give padded WCEC to schedulability analysis

$$WCEC' = WCEC + \max_{1 \leq k \leq s} (\text{headstart}_k)$$

# Dynamic Headstart Accrual

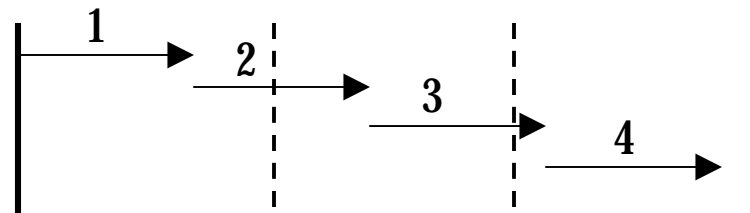
- Harness naturally occurring dynamic slack in simple mode as headstart
- $slack \geq \max_{x \leq k \leq s} (headstart_k)$  à switch to complex mode

# Dynamic Headstart Accrual

simple mode  $\longrightarrow$   
 complex mode  $\cdots\cdots\longrightarrow$

WCEC

**Non-speculative  
 simple mode**



**Successful  
 speculation in  
 complex mode**



WCEC

accrued slack >  $\max(\text{headstart}_{2,3,4})$  ?  
**NO**

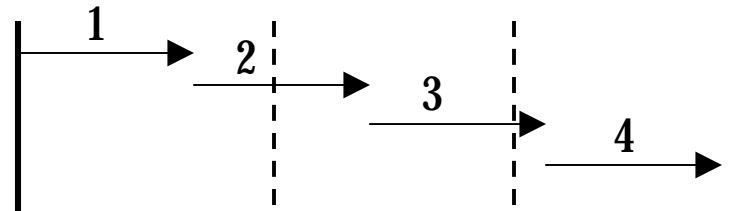


# Dynamic Headstart Accrual

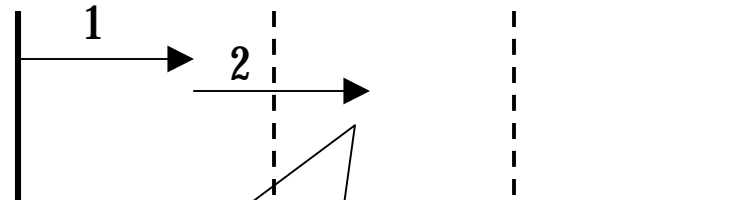
simple mode  $\longrightarrow$   
 complex mode  $\cdots\cdots\longrightarrow$

WCEC

**Non-speculative  
 simple mode**



**Successful  
 speculation in  
 complex mode**

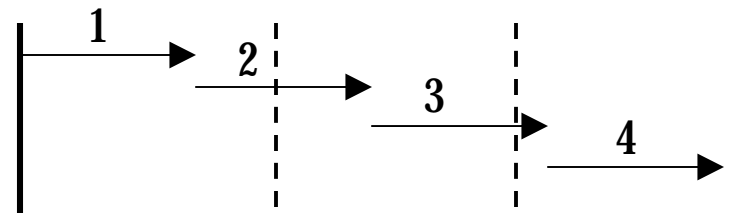


accrued slack  $>$   $\max(\text{headstart}_{3,4})$  ?  
**YES**

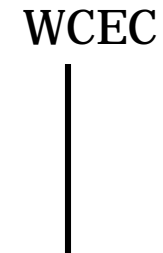
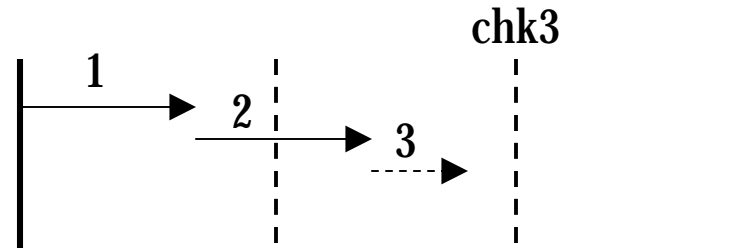
# Dynamic Headstart Accrual

simple mode  $\longrightarrow$   
 complex mode  $\dashrightarrow$   
 WCEC

**Non-speculative  
 simple mode**



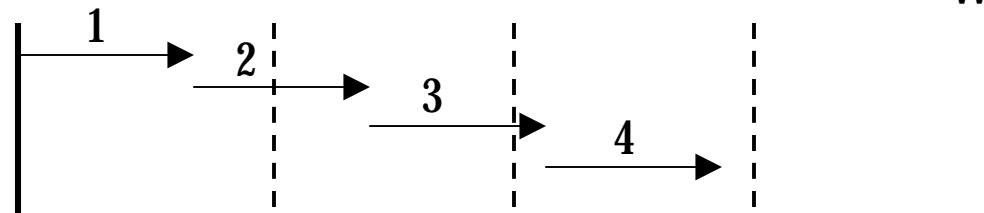
**Successful  
 speculation in  
 complex mode**



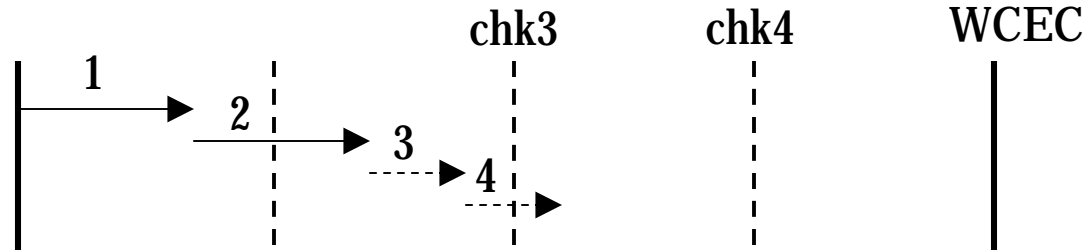
# Dynamic Headstart Accrual

simple mode  $\longrightarrow$   
 complex mode  $\cdots\cdots\longrightarrow$   
 WCEC

**Non-speculative  
 simple mode**



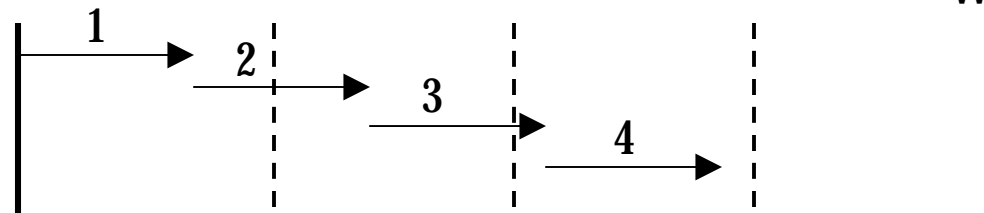
**Successful  
 speculation in  
 complex mode**



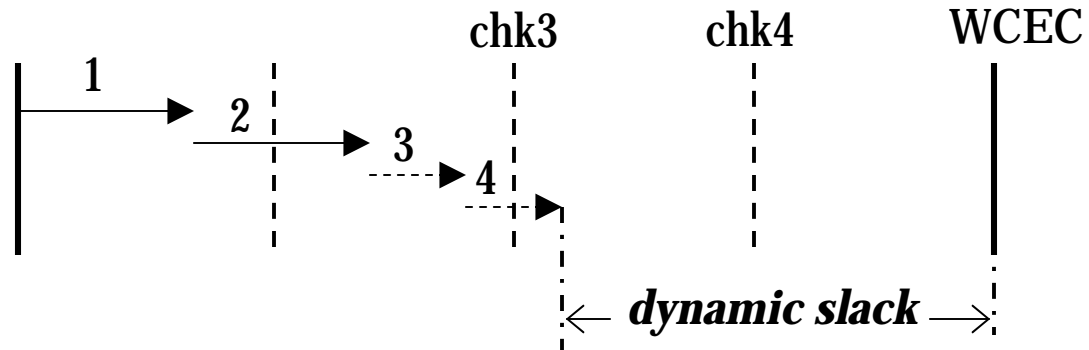
# Dynamic Headstart Accrual

simple mode  $\longrightarrow$   
 complex mode  $\cdots\cdots\longrightarrow$   
 WCEC

**Non-speculative  
 simple mode**



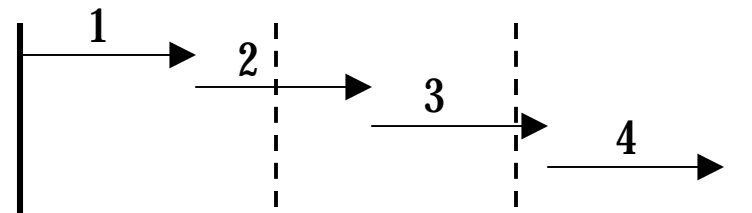
**Successful  
 speculation in  
 complex mode**



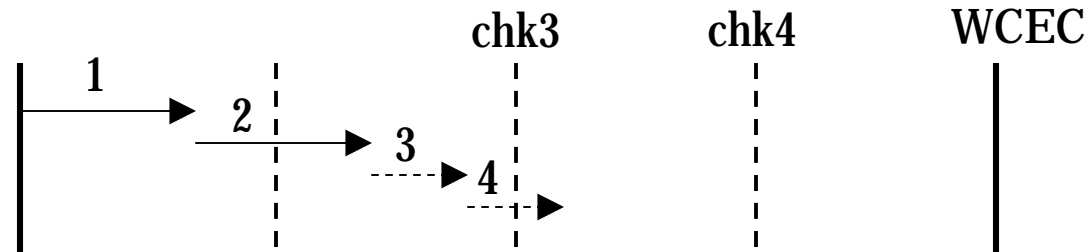
# Dynamic Headstart Accrual

simple mode  $\longrightarrow$   
 complex mode  $\cdots\cdots\longrightarrow$   
 WCEC

**Non-speculative  
simple mode**



**Successful  
speculation in  
complex mode**

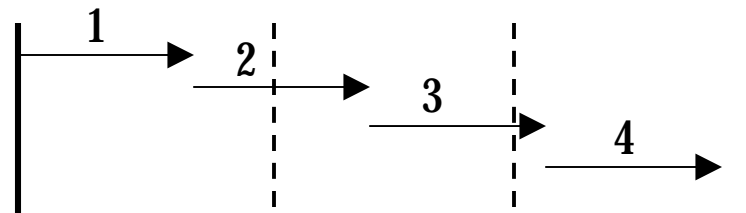


- **First simple mode, then complex mode**
- **No explicit headstart padding**

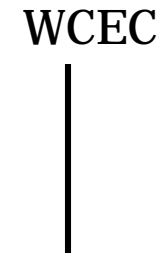
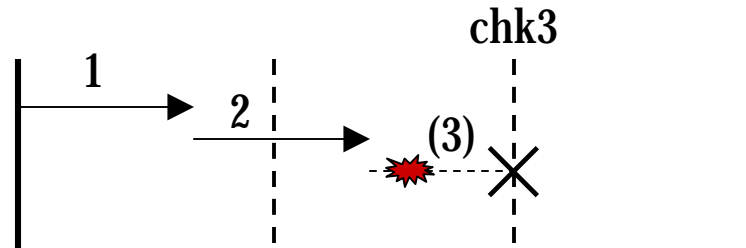
# Dynamic Headstart Accrual

simple mode  $\longrightarrow$   
 complex mode  $\cdots\cdots\longrightarrow$   
 WCEC

**Non-speculative  
 simple mode**



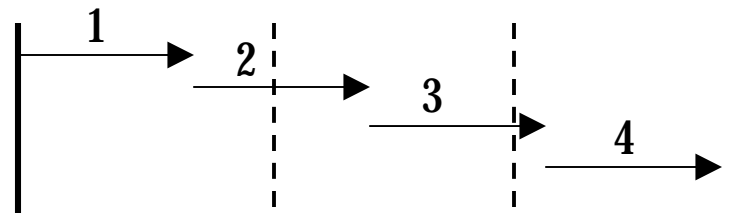
**Misspeculation  
 in complex mode**



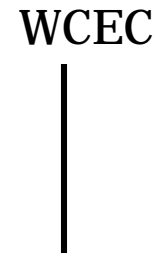
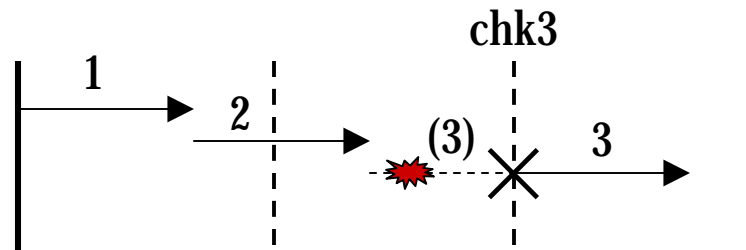
# Dynamic Headstart Accrual

simple mode  $\longrightarrow$   
 complex mode  $\dashrightarrow$   
 WCEC

**Non-speculative  
 simple mode**



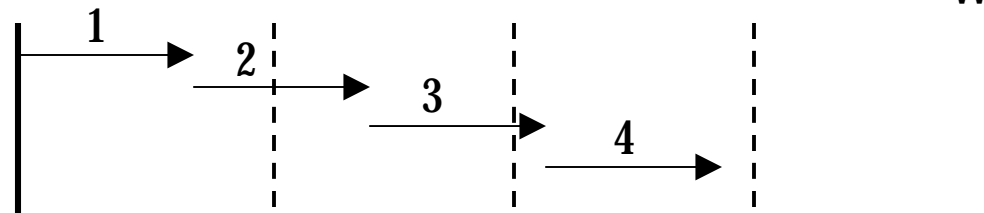
**Misspeculation  
 in complex mode**



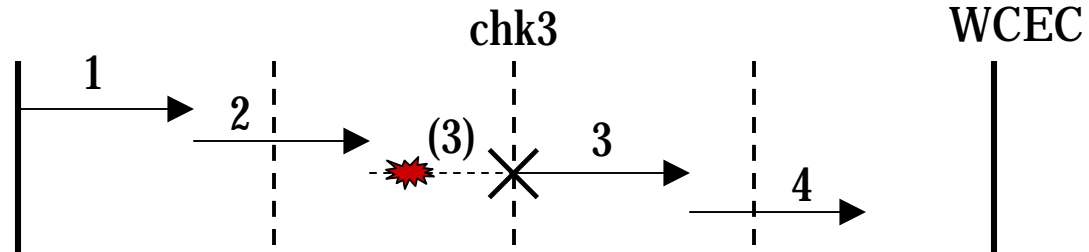
# Dynamic Headstart Accrual

simple mode  $\longrightarrow$   
 complex mode  $\cdots\cdots\longrightarrow$   
 WCEC

**Non-speculative  
 simple mode**



**Misspeculation  
 in complex mode**

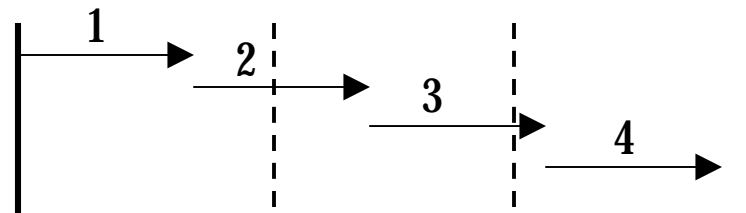




# Dynamic Headstart Accrual

simple mode  $\longrightarrow$   
 complex mode  $\dashrightarrow$   
 WCEC

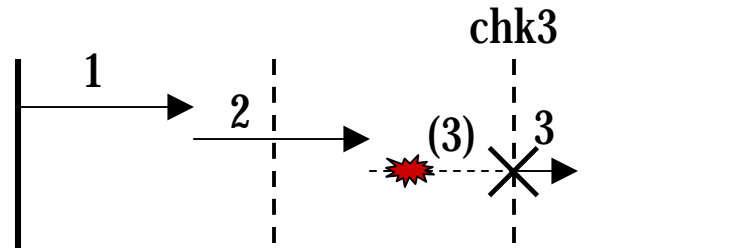
**Non-speculative  
 simple mode**



WCEC



**Flexible: fluidly  
 switch between  
 simple and  
 complex**



WCEC

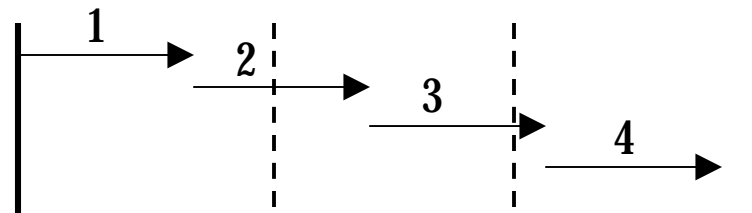


# Dynamic Headstart Accrual

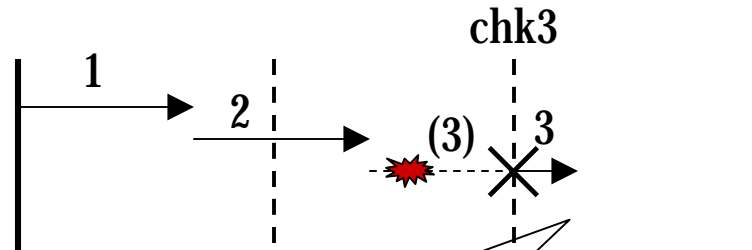
simple mode  $\longrightarrow$   
 complex mode  $\dashrightarrow$

WCEC

**Non-speculative  
 simple mode**



**Flexible: fluidly  
 switch between  
 simple and  
 complex**



WCEC

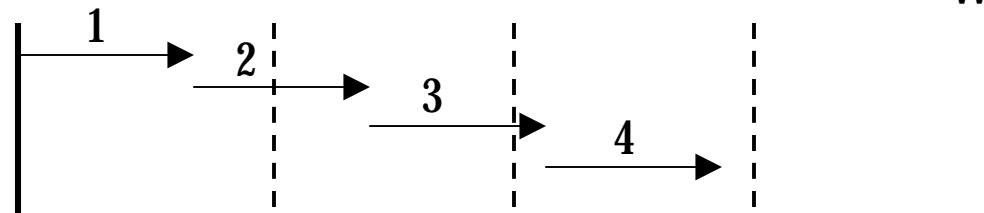


accrued slack > headstart<sub>4</sub> ?  
**YES**

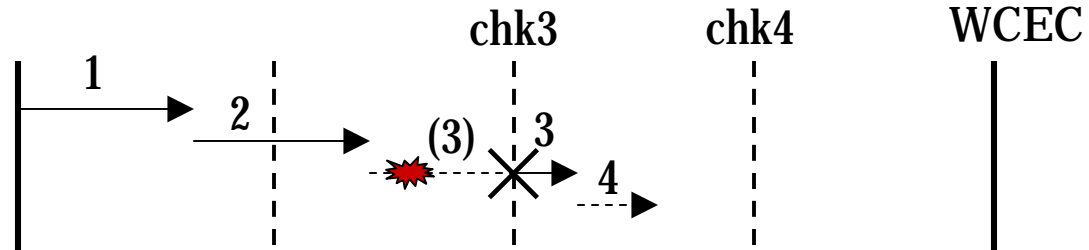
# Dynamic Headstart Accrual

simple mode  $\longrightarrow$   
 complex mode  $\dashrightarrow$   
 WCEC

**Non-speculative  
 simple mode**



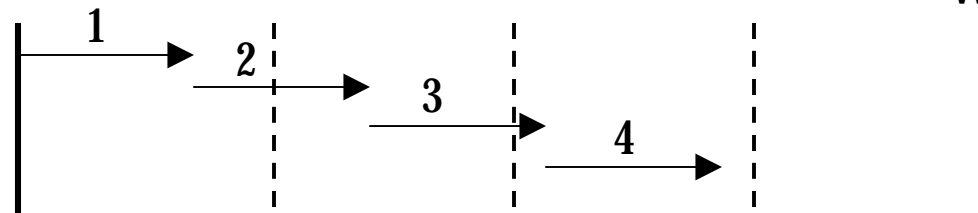
**Flexible: fluidly  
 switch between  
 simple and  
 complex**



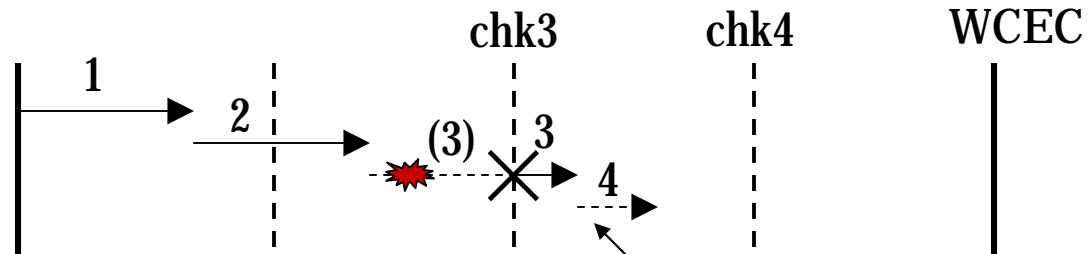
# Dynamic Headstart Accrual

simple mode  $\longrightarrow$   
 complex mode  $\dashrightarrow$   
 WCEC

**Non-speculative  
 simple mode**



**Flexible: fluidly  
 switch between  
 simple and  
 complex**



*Re-enable speculation  
 after missed checkpoint*

# Explicit Padding vs. Dynamic Headstart Accrual

- **Explicit padding**
  - + Guaranteed speculation
  - Inflated WCETs → Unschedulable task-sets
- **Dynamic headstart accrual**
  - + Schedulability unaffected
  - + Flexible switching
  - Dependent on dynamic slack in simple mode

# VISA in Multi-Tasking Systems

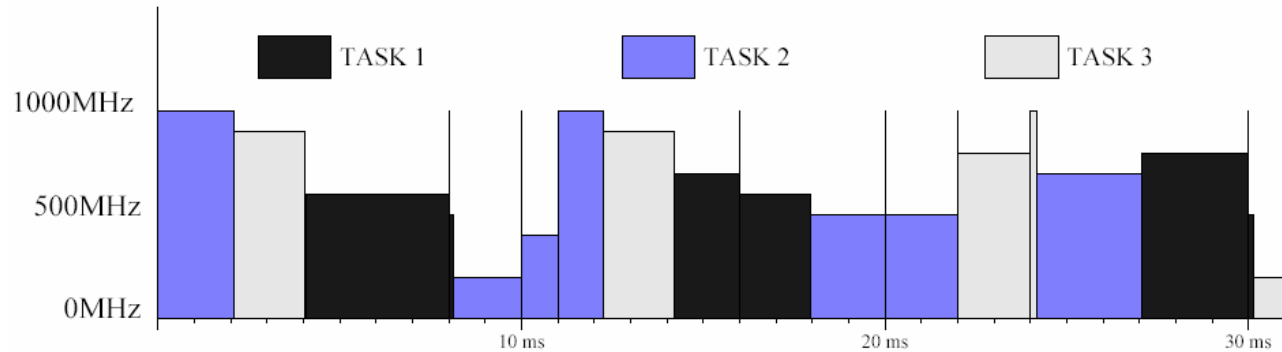
- Gauging mechanism (watchdog counter) disrupted
- Adapt for multi-tasking
  - Interruption → save watchdog counter
  - Resumption → restore watchdog counter

## Easy Integration in Multi-Tasking Systems

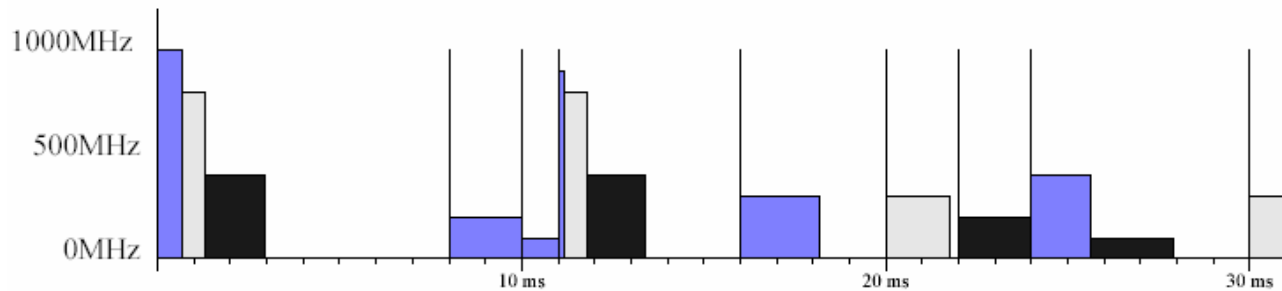
- System software components depend on WCET
  - EDF scheduler, DVS scheduler, etc.
- **VISA preserves WCET abstraction**
- We demonstrate VISA in a hard-real-time system with Look-Ahead EDF-DVS [Pillai&Shin'01]

# Look-Ahead EDF-DVS in VISA

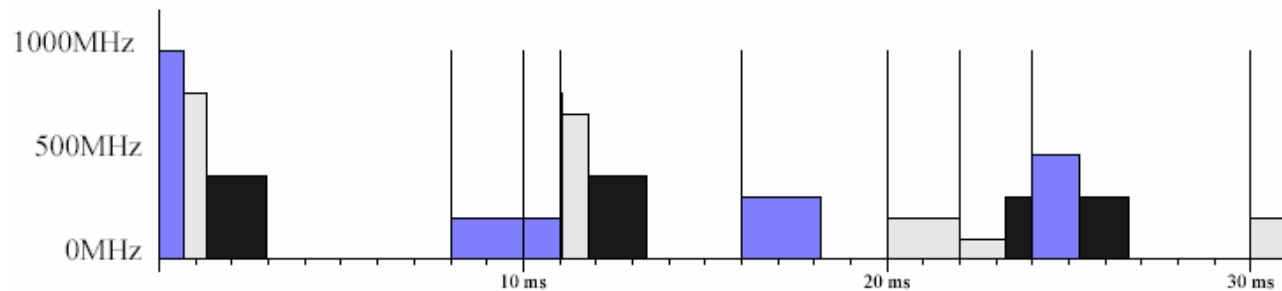
*Simple processor*



*VISA  
(Explicit padding)*



*VISA  
(Dynamic  
headstart accrual)*

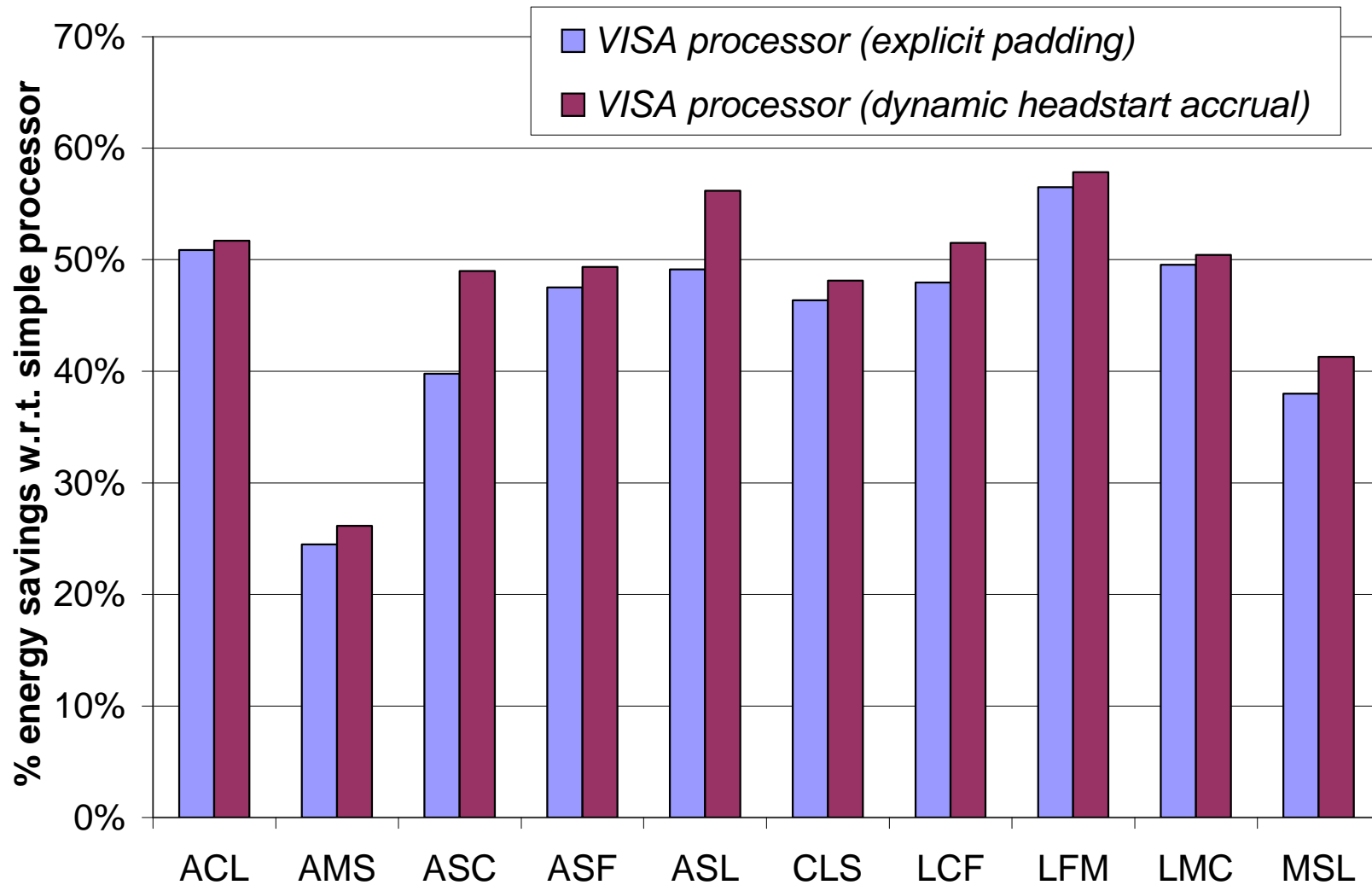




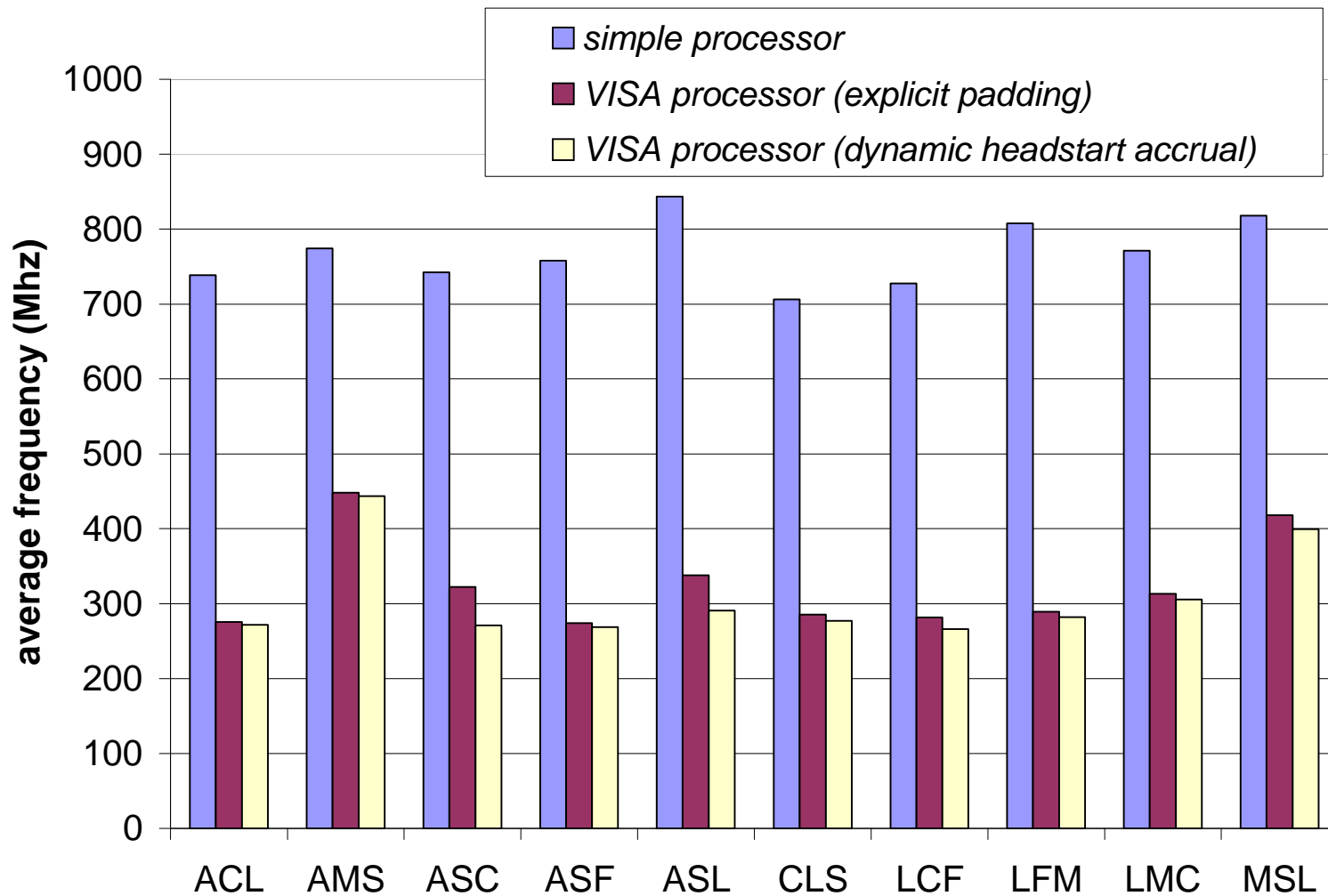
# Experimental Methodology

- Cycle-accurate microarchitecture simulator
- Watch power models to measure power/energy  
[Brooks00]
- 6 C-lab real-time benchmarks

# Energy Savings



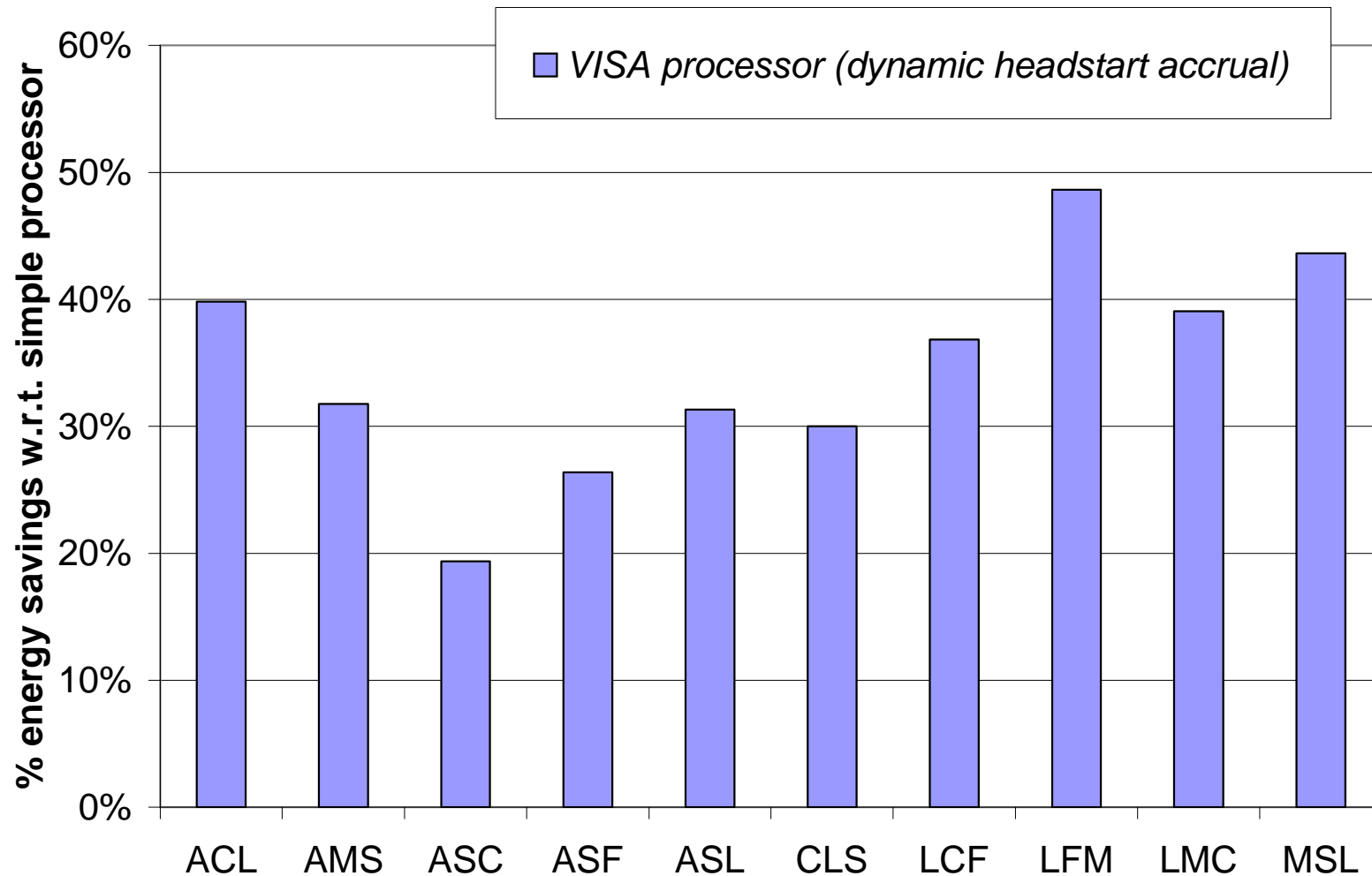
# Average Frequencies



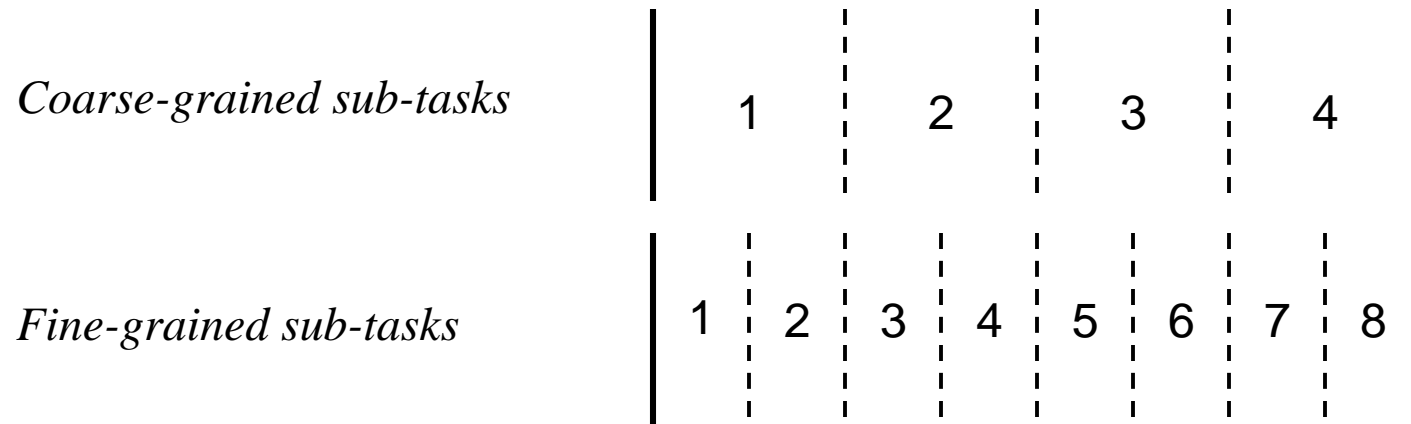
# High Utilization Task-sets

- Worst-case utilization (unpadded WCETs) = 1.0
  - Cannot use explicit padding  $\Rightarrow$  task-set unschedulable
  - Dynamic headstart accrual works!

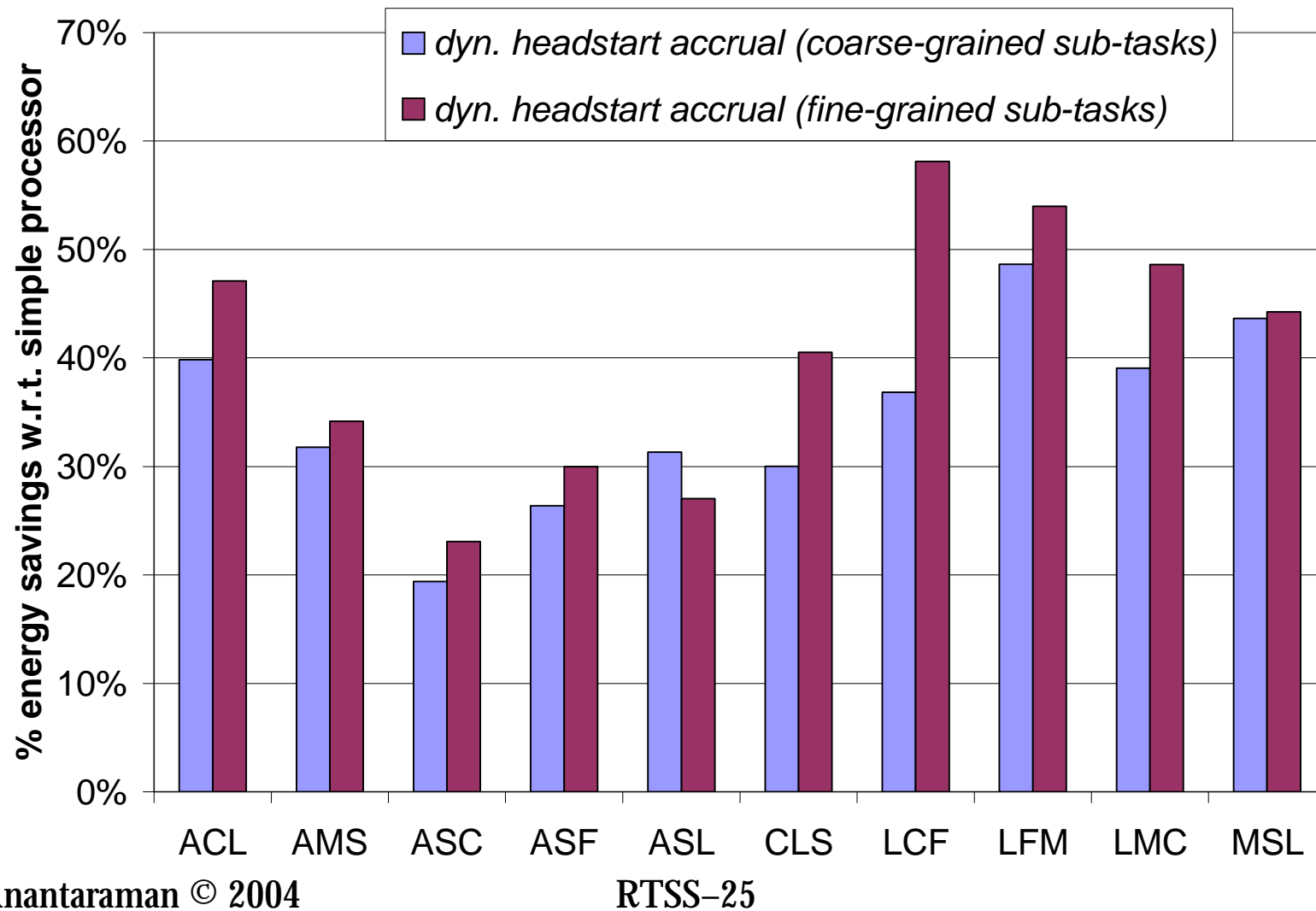
# Energy Savings (U = 1)



# Coarse-grained vs. fine-grained sub-tasks



# Fine-grained vs. coarse-grained sub-tasks



# Summary

- VISA enables use of complex processors in safe real-time systems
- Headstart calculation
- Novel zero-overhead VISA speculation technique
  - *dynamic headstart accrual*
- VISA extended to multi-tasking systems
- 19% – 58% energy savings with respect to explicitly-safe simple processor



# Questions?